Defensive Programming

# Before Class

1. Zapoznaj się z materiałem dotyczącym programowania defensywnego - lekcje od 40 do 48 na kanale „Introduction to Computer Science and Programming Using Python (MIT)”.
2. Zapoznaj się z materiałami dostępnymi w witrynie w3schools.com dotyczącymi obsługi wyjątków.
3. Zapoznaj się z materiałami dotyczącymi błędów oraz obsługi i zgłaszania wyjątków w języku Python

<https://docs.python.org/3/tutorial/errors.html>  
<https://wiki.python.org/moin/UsingAssertionsEffectively>

# During Class

## Assertions

1. Napisz program, który wyznaczy rezultat dzielenia dwóch liczb całkowitych. Zastosuj asercję dla sprawdzenia, czy mianownik jest różny od zera. Sprawdź działanie programu. Zmień wartość b przypisując 0 i zobacz, jaki będzie rezultat działania programu.

a = 5  
b = 3  
assert b!=0  
print(f'{a}/{b} = {a/b}')

1. W poprzednim programie dokonaj również sprawdzenia typu wartości zmiennych a i b (wartości całkowite). Wykorzystaj funkcję type: type(a) == int
2. Napisz program obliczający wskaźnik BMI. Dane dotyczące wzrostu i wagi odczytaj z klawiatury. Stosując asercje sprawdź, czy wzrost wyrażony w cm jest liczbą całkowitą i mieści się w przedziale <150;220>, natomiast waga wyrażona w kg jest liczbą rzeczywistą i mieści się w przedziale <40.0; 150.0>

## Exception handling

1. Napisz program, który obliczy i wyświetli pierwiastek kwadratowy liczby wprowadzonej z klawiatury. Zastosuj obsługę wyjątków dla sprawdzenia poprawności wprowadzanych danych (liczba >= 0). Sprawdź działanie programu wprowadzając poprawną liczbę, a także wprowadzając niepoprawną wartość (liczba < 0 lub wartość innego typu, niż liczba). Zaobserwuj, jaki będzie rezultat działania programu w każdym z tych przypadków.

import math  
  
try:  
 number = float(input('Enter any number: '))  
 print (f'sqrt({number}) = {math.sqrt(number)}' )  
except:  
 print('Please enter a number greater than 0')

1. W grupach 2-3 osobowych, zmodyfikuj powyższy program, aby działał on do momentu wprowadzenia poprawnej liczby większej od 0. Zastosuj obsługę wyjątków oraz instrukcję pętli.
2. Napisz program, który wyświetli zawartość pliku NoEducation.txt. Uwzględnij sytuację wyjątkową (brak pliku lub jego niepoprawne otwarcie). Zastosuj obsługę wyjątków
3. IOError

## Raising exceptions

# After Class

1. Zadanie na obsługę wyjątku już istniejącego np. niezgodny typ lub brak pliku do otwarcia
2. Zadanie na zgłoszenie wyjątku